

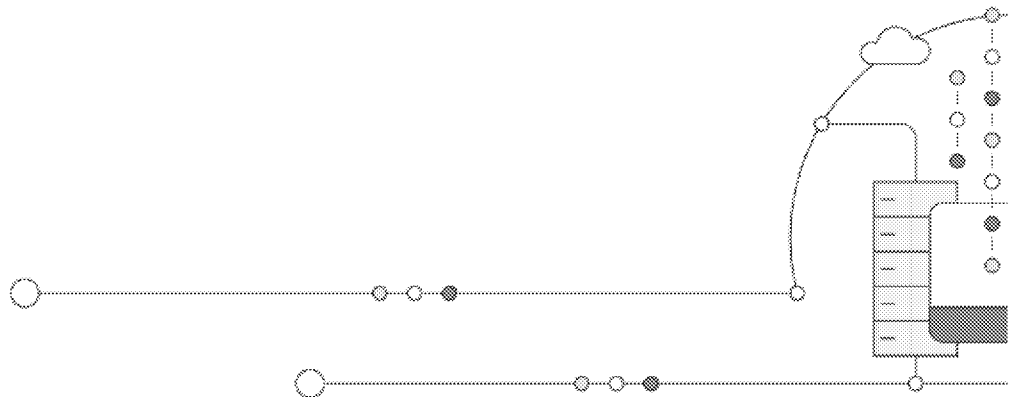
EXHIBIT 58

Home / Topics / Kubernetes

What is Kubernetes?

Explore IBM's Kubernetes solution →

Sign up for cloud updates →



✕ Close

Hello! How can we help you?

Overview

Background: Containers, Docker and Kubernetes

What does Kubernetes do?

Kubernetes architecture and components

The Kubernetes ecosystem

Kubernetes use cases

Kubernetes tutorials

Related solutions

Resources

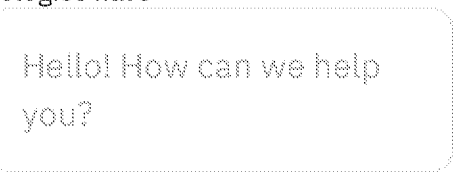
Take the next step

Published: 11 March 2024
Contributors: Stephanie Susnjara, Ian Smalley

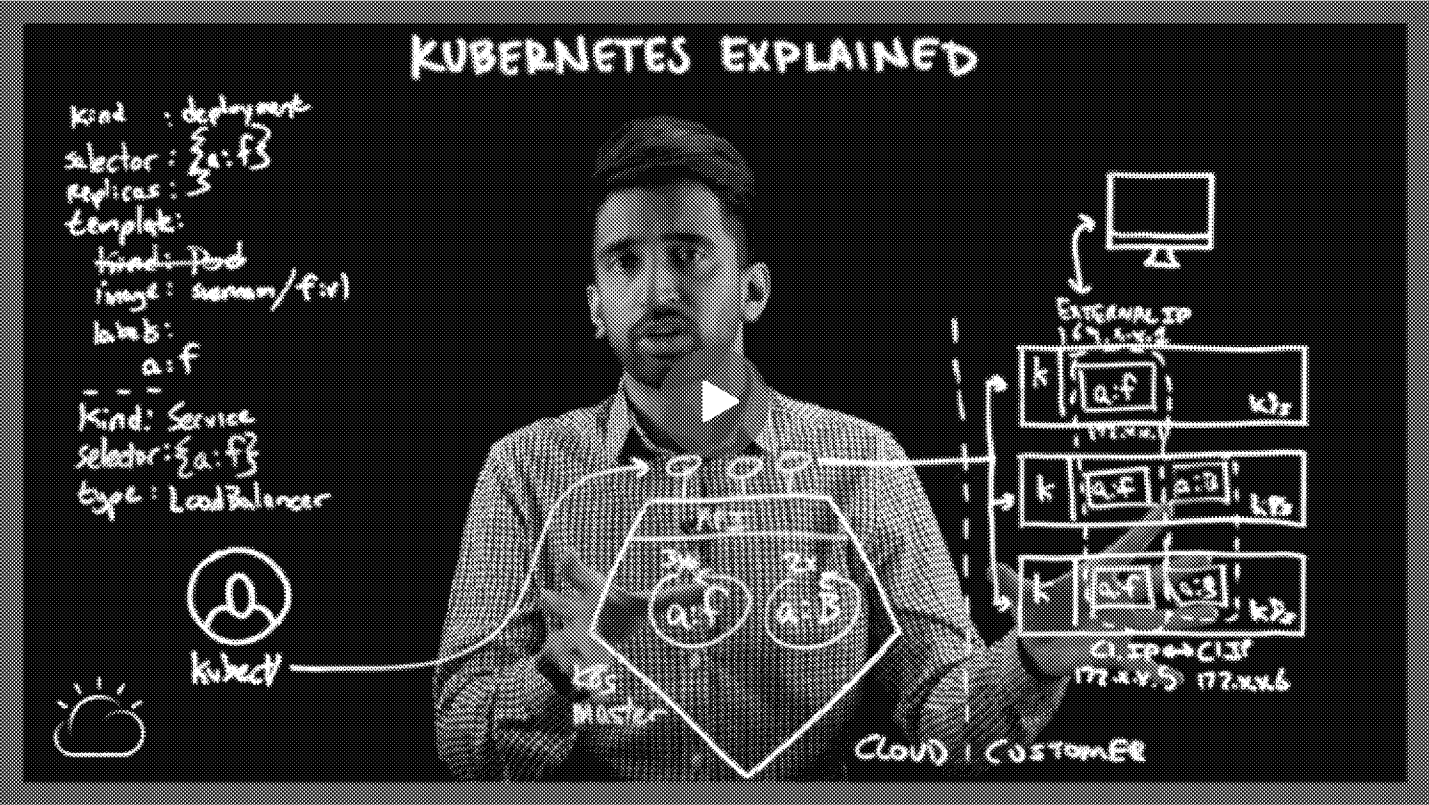
What is Kubernetes?

Kubernetes, also known as k8s or kube, is an open source container orchestration platform for scheduling and automating the deployment, management and scaling of containerized applications.

Today, Kubernetes and the broader ecosystem of container-related technologies have merged to form the building blocks of modern cloud infrastructure. This enables organizations to deliver a highly productive hybrid multicloud environment to perform complex tasks surrounding infrastructure and operations. Kubernetes supports cloud-native development by enabling a build-once-and-deploy approach to building applications.



The word Kubernetes originates from Greek, meaning *helmsman* or *pilot*, hence the helm in the Kubernetes logo (link resides outside of ibm.com).



Kubernetes Explained (10:59)

Guide

Achieve workplace flexibility with DaaS

Read how desktop as a service (DaaS) enables enterprises to achieve the same level of performance and security as deploying the applications on-premises.



Hello! How can we help you?

Related content

Register for the guide on hybrid cloud



Background: Containers, Docker and Kubernetes

What are containers?

Containers are lightweight, executable application components that combine source code with all the operating system (OS) libraries and dependencies required to run the code in any environment.

Containers take advantage of a form of OS virtualization that allows multiple applications to share a single instance of an OS by isolating processes and controlling the amount of CPU, memory and disk those processes can access. Because they are smaller, more resource-efficient and more portable than virtual machines (VMs), containers have become the de facto compute units of modern cloud-native applications. Containers are also more resource-efficient. They allow you to run more applications on fewer machines (virtual servers and physical servers) with fewer OS instances.

Since containers can run consistently anywhere, they have become critical to the underlying architecture that supports hybrid multicloud environments, the combination of on-premises, private cloud, public cloud and more than one cloud service from more than one cloud vendor.

What is Docker?

Docker is the most popular tool for creating and running Linux® container forms of containers were introduced decades ago (with technologies such as Jails and AIX Workload Partitions), containers were democratized in 2014 when Docker brought them to the masses with a new developer-friendly and cloud-friendly implementation.

Hello! How can we help you?

Docker began as an open source project, but today, it also refers to Docker Inc., the company that produces Docker—a commercial container toolkit that builds on the open source project and contributes those improvements back to the open source community.

Docker was built on traditional Linux container technology but enables more granular virtualization of Linux kernel processes and adds features to make containers more accessible for developers to build, deploy, manage and secure.

While alternative container runtime platforms exist today like Open Container Initiative (OCI), CoreOS and Canonical (Ubuntu) LXD, Docker is the dominant choice. Moreover, Docker has become synonymous with containers and is sometimes mistaken as a competitor to complimentary technologies like Kubernetes.

Today, Docker and Kubernetes are the leading containerization tools, with Docker dominating 82% of the market and Kubernetes controlling 11.52% market share in 2024 (link resides outside ibm.com).

Container orchestration with Kubernetes

As containers proliferated, today, an organization might have hundreds or thousands of them. Operations teams are needed to schedule and automate container deployment, networking, scalability and availability. Enter container orchestration.

Based on Borg, Google's internal container orchestration platform, Kubernetes was introduced to the public as an open source tool in 2014, with Microsoft, Red Hat®, IBM and other major tech players signing on as early members of the Kubernetes community. In 2015, Google donated Kubernetes to the Cloud Native Computing Foundation (CNCF) (link resides outside ibm.com), the open source, vendor-neutral hub of cloud-native computing.

Kubernetes became the CNCF's first hosted project in March 2016. Since then, Kubernetes has become the most widely used container orchestration tool for running container-based workloads worldwide. In a CNCF report (link resides outside ibm.com), Kubernetes is the second largest open source project in the world (after Linux) and the primary container orchestration tool for 71% of Fortune 100 companies.

In 2018, Kubernetes was the CNCF's first graduate project, becoming one of the fastest growing open source projects in history. While other container orchestration tools, most notably Docker Swarm and Apache Mesos, gained some traction early on, Kubernetes quickly became the most widely adopted.

Hello! How can we help you?

Since Kubernetes joined the CNCF in 2016, the number of contributors has grown to 8,012, a 996% increase (link resides outside ibm.com). As of this writing, contributors have added over 123,000 commits to the Kubernetes repository on GitHub (link resides outside ibm.com).

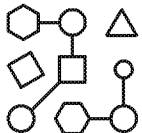
What does Kubernetes do?

Kubernetes schedules and automates container-related tasks throughout the application lifecycle, including the following.



Deployment

Deploy a specified number of containers to a specified host and keep them running in a wanted state.



Rollouts

A rollout is a change to a deployment. Kubernetes lets you initiate, pause, resume or roll back rollouts.



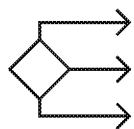
Service discovery

Kubernetes can automatically expose a container to the internet or to other containers by using a domain name system (DNS) name or IP address.



Storage provisioning

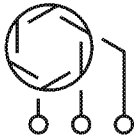
Set Kubernetes to mount persistent local or cloud storage for your containers as needed.



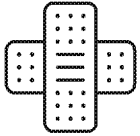
Load balancing

Based on CPU usage or custom metrics, K the workload across the network to maint

Hello! How can we help you?

**Autoscaling**

When traffic spikes, Kubernetes autoscaling can spin up new clusters as needed to handle the additional workload.

**Self-healing for high availability**

When a container fails, Kubernetes can restart or replace it automatically to prevent downtime. It can also take down containers that don't meet your health check requirements.

Kubernetes architecture and components

Deploying Kubernetes involves clusters, the building blocks of Kubernetes architecture. Clusters are made up of nodes, each representing a single compute host, either a physical machine (bare metal server) or a VM.

Kubernetes architecture consists of two main parts: the control plane components and the components that manage individual nodes.

A node consists of pods. These are groups of containers that share the same computing resources and the same network. They are also the unit of scalability in Kubernetes. If a container in a pod is gaining more traffic than it can handle, Kubernetes will replicate the pod to other nodes in the cluster.

The control plane automatically handles scheduling the pods across the nodes in a cluster.

Hello! How can we help you?

Control plane components

Each cluster has a master node that handles the cluster's control plane. The master node runs a scheduler service that automates when and where the containers are deployed based on developer-set deployment requirements and available computing capacity.

The main components in a Kubernetes cluster are the kube-apiserver, etcd, kube-scheduler, kube-controller-manager and cloud-controller-manager:

API server: The application programming interface (API) server in Kubernetes exposes the Kubernetes API (the interface used to manage, create and configure Kubernetes clusters) and serves as the entry point for all commands and queries.

etcd: The etcd is an open source, distributed key-value store used to hold and manage the critical information that distributed systems need to keep running. In Kubernetes, the etcd manages the configuration data, state data and metadata.

Scheduler: This component tracks newly created pods and selects nodes for them to run on. The scheduler considers resource availability and allocation restraints, hardware and software requirements, and more.

Controller-manager: A set of built-in controllers, the Kubernetes controller-manager runs a control loop that monitors the shared state of the cluster and communicates with the API server to manage resources, pods or service endpoints. The controller-manager consists of separate processes that are bundled together to reduce complexity and run in one process.

Cloud-controller-manager: This component is similar in function to the controller-manager link. It links to a cloud provider's API and separates the components that interact with that cloud platform from those that only interact within the cluster.

Node components

Worker nodes are responsible for deploying, running and managing containerized applications:

Kubelet: Kubelet is a software agent that receives and runs orders from the master node and helps to ensure that containers run in a pod.

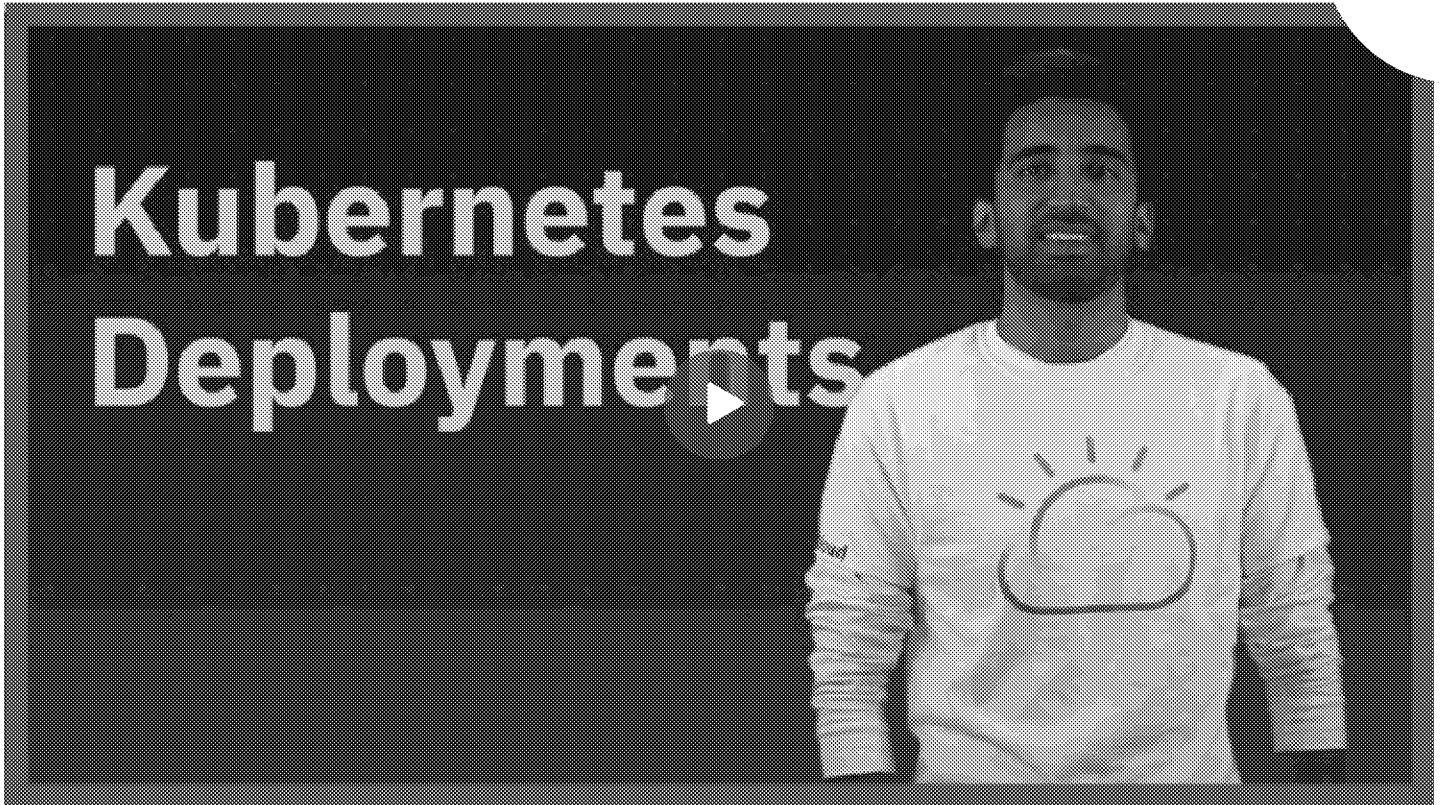
Kube-proxy: Installed on every node in a cluster, the kube-proxy maintains the host and monitors changes in services and pods.

Hello! How can we help you?

Other Kubernetes concepts and terminology

- **ReplicaSet:** A ReplicaSet maintains a stable set of replica pods for specific workloads.
- **Deployment:** The deployment controls the creation and state of the containerized application and keeps it running. It specifies how many replicas of a pod should run on the cluster. If a pod fails, the deployment creates a new one.
- **Kubecttl:** Developers manage cluster operations by using kubecttl, a developer tool consisting of a command-line interface (CLI) that communicates directly with the Kubernetes API.
- **DaemonSets:** DaemonSets are responsible for helping to ensure that a pod is created on every single node in the cluster.
- **Add-ons:** Kubernetes add-ons extend functions and include Cluster DNS (a DNS server that provides DNS records to Kubernetes.), Web UI (a Kubernetes dashboard for managing a cluster), and more.
- **Service:** A Kubernetes service is an abstraction layer that defines a logical set of pods and how to access them. A service exposes a network application running on one or more pods in a cluster. It provides an abstract way to load balance pods.

Hello! How can we help you?



Kubernetes Deployments: Get Started Fast (3:55)

The Kubernetes ecosystem

Today, there are over 90 certified Kubernetes offerings (link resides outside ibm.com), including enterprise-grade management platforms that provide tools, upgrades and add-on capabilities that accelerate the development and delivery of containerized applications.

Managed Kubernetes services

While Kubernetes is the technology of choice for orchestrating container applications, it depends on other components, ranging from networking, balancing, storage, continuous integration and continuous delivery (CI/C be fully functional.

Hello! How can we help you?

While self-hosting a Kubernetes cluster in a cloud-based environment is possible, setup and management can be complex for an enterprise organization. This is where managed Kubernetes services come in.

With managed Kubernetes services, the provider typically manages the Kubernetes control plane components. The managed service provider helps automate routine processes for updates, load balancing, scaling and monitoring. For example, Red Hat® OpenShift® is a Kubernetes service that can be deployed in any environment and on all major public clouds including Amazon Web Services (AWS), Microsoft Azure, Google Cloud and IBM Cloud®. Many cloud providers also offer their own managed Kubernetes services.

Kubernetes monitoring tools

Kubernetes monitoring refers to collecting and analyzing data related to the health, performance and cost characteristics of containerized applications running inside a Kubernetes cluster.

Monitoring Kubernetes clusters allows administrators and users to track uptime, usage of cluster resources and the interaction between cluster components. Monitoring helps to quickly identify issues like insufficient resources, failures and nodes that can't join the cluster. Today's Kubernetes monitoring solutions include automated tools for application performance management (APM), observability, application resource management (ARM) and more.

Istio service mesh

Kubernetes can deploy and scale pods, but it can't manage or automate routing between them and doesn't provide any tools to monitor, secure or debug these connections.

As the number of containers in a cluster grows, the number of possible connection paths between them escalates exponentially. For example, 2 containers have 2 potential connections, but 10 pods have 90, creating a potential configuration and management nightmare.

Istio, a configurable, open source service mesh layer, provides a solution for monitoring and securing containers in a Kubernetes cluster. Other significant benefits include capabilities for improved debugging and a dashboard that DevOps teams and

Hello! How can we help you?

administrators can use to monitor latency, time-in-service errors and other characteristics of the connections between containers.

Knative and serverless computing

Knative (pronounced 'kay-native') is an open source platform that provides an easy onramp to serverless computing, the cloud computing application development and execution model that enables developers to build and run application code without provisioning or managing servers or backend infrastructure.

Instead of deploying an ongoing instance of code that sits idle while waiting for requests, serverless brings up the code as needed, scaling it up or down as demand fluctuates, and then takes down the code when not in use. Serverless prevents wasted computing capacity and power and reduces costs because you only pay to run the code when it's running.

Tekton

Tekton is an open source, vendor-neutral framework for creating continuous integration and delivery (CI/CD) systems governed by the Continuous Delivery Foundation (CDF) (link resides outside ibm.com).

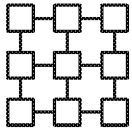
As a Kubernetes framework, Tekton helps modernize continuous delivery by providing industry specifications for pipelines, workflows and other building blocks, making deployment across multiple cloud providers or hybrid environments faster and easier.

It's worth noting that Tekton is the successor to Knative Build, which is still supported in some Knative distributions. Tekton pipelines have become the standard for building container images and deploying them in a container registry in a Kubernetes environment.

Hello! How can we help you?

Kubernetes use cases

Enterprise organizations use Kubernetes to support the following use cases that all play a crucial role in comprising modern IT infrastructure.



Microservices architecture or cloud-native development

Cloud native is a software development approach for building, deploying and managing cloud-based applications. The major benefit of cloud-native is that it allows DevOps and other teams to code once and deploy on any cloud infrastructure from any cloud service provider.

This modern development process relies on microservices, an approach where a single application is composed of many loosely coupled and independently deployable smaller components or services, which are deployed in containers managed by Kubernetes.

Kubernetes helps ensure that each microservice has the resources it needs to run effectively while also minimizing the operational overhead associated with manually managing multiple containers.



Hybrid multicloud environments

Hybrid cloud combines and unifies public cloud, private cloud and on-premises data center infrastructure to create a single, flexible, cost-optimal IT infrastructure.

Today, hybrid cloud has merged with multicloud, public cloud services from more than one cloud vendor, to create a hybrid multicloud environment.

A hybrid multicloud approach creates greater flexibility and reduces an organization's dependency on one vendor, preventing vendor lock-in. Since Kubernetes creates the foundation for cloud-native development, it's key to hybrid multicloud adoption.



Applications at scale

Kubernetes supports large-scale cloud app deployment with autoscaling. This process allows applications to scale up or down, adjusting to demand changes automatically, with speed, efficiency and minimal downtime.

The elastic scalability of Kubernetes deployment allows applications to be added or removed based on changes in user demand or website traffic.

Hello! How can we help you?

**Application modernization**

Kubernetes provides the modern cloud platform needed to support application modernization, migrating and transforming monolithic legacy applications into cloud applications built on microservices architecture.

**DevOps practices**

Automation is at the core of DevOps, which speeds the delivery of higher-quality software by combining and automating the work of software development and IT operations teams.

Kubernetes helps DevOps teams build and update apps rapidly by automating the configuration and deployment of applications.

**Artificial intelligence (AI) and machine learning (ML)**

The ML models and large language models (LLM) that support AI include components that would be difficult and time-consuming to manage separately. By automating configuration, deployment and scalability across cloud environments, Kubernetes helps provide the agility and flexibility needed to train, test and deploy these complex models.

Kubernetes tutorials

If you're ready to start working with Kubernetes or looking to build your skills with Kubernetes and Kubernetes ecosystem tools, try one of these tutorials:

- Kubernetes tutorials: Free hands-on labs with certification
- IBM Developer: Kubernetes tutorials
- Kubernetes Tutorials: 5 ways to get you building fast
- 8 Kubernetes tips and tricks
- Debug and log your Kubernetes applications

Hello! How can we help you?

Related solutions

Red Hat OpenShift on IBM Cloud

With Red Hat OpenShift on IBM Cloud, OpenShift developers have a fast and secure way to containerize and deploy enterprise workloads in Kubernetes clusters.

Explore Red Hat OpenShift →

IBM Cloud® Kubernetes Service

Deploy secure, highly available clusters in a native Kubernetes experience.

Explore IBM Cloud Kubernetes Service →

IBM Cloud Pak® for Applications

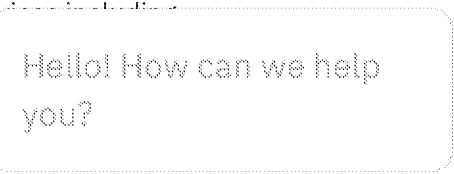
Deliver an application landscape that transforms with your business. IBM Cloud Pak for Applications provides the ultimate flexibility whether you're building new cloud-native services and applications, or refactoring or re-platforming existing applications.

Explore IBM Cloud Pak for Applications →

IBM Cloud® Satellite

Deploy and run apps consistently across on-premises, edge computing and public cloud environments from any cloud vendor, by using a common set of cloud service toolchains, databases and AI.

Explore IBM Cloud Satellite Solutions →



IBM Cloud® Code Engine

A fully managed serverless platform, IBM Cloud Code Engine lets you run your container, application code or batch job on a fully managed container runtime.

Explore Code Engine →

Resources

E-book

Containers in the enterprise

IBM Research® documents the surging momentum of container and Kubernetes adoption:

Read book

Learn

Flexible, resilient, secure IT for you

Part of a hybrid cloud strategy that lets you build and manage workloads from anywhere

→

Article

What is serverless?

Serverless is a cloud application development and execution model that lets developers build and run code without managing servers

Tutorial

YAML basics in Kubernetes

Explore an example of how a YAML file is used in Kubernetes.

Learn more →

Article

What are containers?

Containers are executable units of software in which application code is packaged along with its libraries and dependencies,

Article

What is Docker?

Docker is an open source platform that enables developers to build, deploy, run, update and manage containers, standard

Hello! How can we help you?

Take the next step

Red Hat OpenShift on IBM Cloud offers developers a fast and secure way to containerize and deploy enterprise workloads in Kubernetes clusters. Offload tedious and repetitive tasks involving security management, compliance management, deployment management and ongoing lifecycle management.

Explore Red Hat OpenShift on IBM Cloud

→

Start for free

→

environ
ment.

Hello! How can we help
you?